UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/786,843	02/25/2004	Jose German Rivera	200312292-1	2936

22879 7590 01/22/2008
HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
WEI, ZHENG	

ART UNIT	PAPER NUMBER
2192	

NOTIFICATION DATE	DELIVERY MODE
01/22/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
mkraft@hp.com
ipa.mail@hp.com

Office Action Summary	**Application No.** 10/786,843	**Applicant(s)** RIVERA ET AL.	
	Examiner Zheng Wei	**Art Unit** 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _25 October 2007_.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1,4-11,14-21,24-31,34-41 and 44-49_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)□ Claim(s) _1, 4-11, 14-21, 24-31, 34-41 and 44-49_ is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _25 February 2004_ is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

Remarks

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 10/25/2007 has been entered.

2. This office action is in response to the amendment filed on 10/25/2007.

3. Claims 1, 11, 21, 31, 41 and 44 have been amended.

4. The objection to Claim 44 is withdrawn in view of Applicant's amendment.

5. The 112 second rejection to claims 21-30 is withdrawn in view of Applicant further explanation.

6. Claims 1, 4-11, 14-21, 24-31, 34-41 and 44-49 remain pending and have been examined.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 1, 4, 5, 7-11, 14, 15, 17-21, 24, 25, 27-35, 37-41 and 44-49 are rejected under 35 U.S.C. 103(a) as being unpatentable over by Williams (Mickey Williams, Microsoft® Visual C#™ .NET) in view of GNU (The GNU C Library, Section “Explicitly Checking Internal Consistency”) and in further view of PHP (PHP Manual, Section “assert_options”)

Claim 1:

Williams discloses a method for monitoring (debugging and tracing) computer software comprising:

- receiving an assertion from an executing process (see for example, p.11, line 3, “When a DefaultTraceListerner object detects that the Assert method has been called from a server process”), but does not explicitly discloses wherein the executing process is integral to an operating system. However, GNU in the same analogous art of application/operating system error checking discloses using assert method in the operating system and report execution error (see for example, p.1, section “Explicitly Checking Internal Consistency”, first paragraph and p.2 third paragraph, “check for an error return from an operating system function”). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to put the assertion in the operation system code to monitor the error from the system call. One would have been motivated to do so to display all the error information and further help to debug problem as suggested by GNU (see for example, p.1, section “Explicitly Checking Internal Consistency”, first paragraph, “These kinds of checks are helpful in debugging problem...”).

wherein receiving an assertion comprises:

- receiving an assertion request (see for example, p.11, line 3, “When a DefaultTraceListerner object detects that the Assert method has been called from a server process”);

  But neither of them explicitly discloses about recognizing a type for assertion request. However, PHP in the same analogous art of assertion, discloses a assert control option (see for example, p.1, Table 1. Assert Options, “assert.active” and “default value 1” for enabling assert() evaluation and related text). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to assertion control option with assert method in Williams to control monitoring process. One would have been motivated to do so to set the various assert() control options as suggested by PHP (see for example, p.1, “Using assert_options() you may set the various assert() control options...”)
- accepting the assertion request when the determined component has assertion requests enabled (see rejection above for the assert_option is set to 1 (enable assert() evaluation)).

- recording the assertion when the assertion is violated (see for example, p.10-11, figure 9-3 ”Dialog box generated by trace and debug output with the Assert method” and related text, also see p.10, section “Asserting That

Expressions Are True", lines 15-16, The Assert method is used to display an error message when a condition that's expected to evaluates as true evaluates as false."); and

- allowing the executing process to continue execution (see for example, p.10-11, figure 9-3 "Dialog box generated by trace and debug output with the Assert method" and button labeled as "Ignore").

Claim 4:

Williams further discloses the method of claim 1 wherein recording the assertion comprises recording a datum that includes at least one of: type of assertion, sequence number of the assertion, time at which the assertion occurred, identification of processor that produced the assertion, identification of process that produced the assertion, identification of the thread that produced the assertion, text of the assertion, stack trace, source line containing the assertion, and file name of the source containing the code that generated the assertion (see for example, p.10-11, figure 9-3 "Dialog box generated by trace and debug output with the Assert method" and related text, also see p.10, lines 1-2, "As you can see, this dialog box includes call stack information when available. Where debug symbols are available, the stack trace includes file name and line number information.")

Claim 5:

Williams also discloses the method of claim 1 wherein recording the assertion comprises writing information regarding the assertion violation to a computer readable medium (see for example, p.9, lines 13-15, "The .NET Framework includes classes to control trace and debug output message and to write output message to files, streams, and event log.").

Claim 7:

Williams further discloses the method of claim 1 further comprising:

- accepting a command from at least one of a control console and a network connection (see for example.p.203, Figure 9-1. "The build property page for a project, on which new symbols are defined", "TraceDemo Property Pages", "Configuration Properties", also see p.13, example configuration file: *SwitchText.exe.config*); and
- updating an enable condition for an assertion class according to the command (see for example, p.13, line 41-p.14, line 2, "Switches are controlled by adding XML element nodes inside the switches element, Multiple switch objects can be configured through a configuration file by adding additional elements to the switches node." , "BooleanSwitch objects are disabled by default and are enabled if they're assigned a nonzero value in a configuration file.")

Claim 8:

Williams further discloses the method of claim 1 further comprising generating an error report according to the recorded assertion (see for example, p.11, lines 7-17. "The Assert method has three versions", "The most basic version simply accepts an expression that triggers an assertion failure message", "The second version of Assert accepts a second parameter that serves as a short error message describing the assertion violation", "The third version of Assert accepts a third parameter that includes detailed information about the assertion violation")

Claim 9:

Williams also disclose the method of claim 8 further comprising dispatching the error report to a real-time assertion monitor (Visual Studio output window) (see for example, p.11, lines 5-6, "Instead, it writes the output message to the Visual Studio Output window and any other debuggers currently accepting output from the Microsoft Win32 OutputDebugString function.").

Claim 10:

Williams further discloses the method of claim 8 wherein generating an error report comprises: retrieving an assertion violation parameter including at least one of: type of assertion, sequence number of the assertion, time at which the assertion occurred, identification of processor that produced the assertion, identification of process that produced the assertion, identification of the thread that produced the assertion, text of the assertion, stack trace, source line containing the assertion, and file name of the source containing the code that generated the assertion; and generating a report file comprising page description statements according to the assertion parameter (see for example, p.10-11, figure 9-3 "Dialog box generated by trace and debug output with the Assert method" and related text, also see p.10, lines 1-2, "As you can see, this dialog box includes call stack information when available. Where debug symbols are available, the stack trace includes file name and line number information.")

Claims 11, 14, 15, 17-20 and 45:

Claims 11-15, 17-20 and 45 are apparatus version of the claimed method addressed in claims 1-5, 7-10 and 44 above for monitoring computer software, wherein such an apparatus/computer system is deemed to be inherent to produce, such as Figure 9-3 dialog box and word above. Therefore, these claims are also anticipated by Williams.

Claims 41 and 49:

See the rejection in claim 1 above.

Claim 44:

Williams also discloses the method of Claim 1 wherein the assertion request type is one of a group of defined assertion macro names (property) (see for example, p.13, section "Using the BooleanSwitch Class", lines 20-21, "The BooleanSwitch class is used to created simple Switch objects that can be either enable or disabled", also see p.22-24, example code)

Claim 48:

Williams also discloses the apparatus for monitoring computer software of Claim 41 wherein the assertion request type is one of a group of defined assertion macro names (property) (see for example, p.13, section "Using the BooleanSwitch Class", lines 20-21, "The BooleanSwitch class is used to created simple Switch objects that can be either enable or disabled", also see p.22-24, example code).

Claims 21, 24, 25, 27-30 and 46:

Claims 21-25, 27-30 and 46 claim a computer software monitoring system comprising: memory capable of storing instructions; processor capable of executing instructions stored in the memory; and software monitor instruction sequence that, when executed by the processor, minimally causes the processor to: receive an assertion from an executing process, record the assertion, and allow the executing process to continue execution. This is a product version of method claims discussed in claims 1-5 and 7-10 above respectively. It is well known in the computer art that the method can be practiced by the computer system to perform the same functionality. Therefore, these claims are also unpatentable over Williams, GNU and PHP.

Claims 31, 34, 35, 37-40 and 47:

Claims 31-35, 37-40 and 47claim a computer-readable medium having computer-executable instructions for performing a method for monitoring computer software. This is another product version of method claims discussed in claims 1-5 and 7-10 above respectively. It is well known in the computer art that the method can be stored and practiced in the computer-readable medium. Therefore, these claims are also unpatentable over Williams, GNU and PHP.

9. Claims 6, 16, 26 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Williams (Mickey Williams, Microsoft® Visual C#™ .NET) in view of GNU (The GNU C Library, Section "Explicitly Checking Internal Consistency") in further view of PHP (PHP Manual, Section "assert_options") and in further view of Cantrill (Bryan M, Cantrill, US 7,146,473)

Claim 6:

Williams, GNU and PHP disclose the method of claim 1 wherein recording the assertion comprises writing information regarding the assertion violation to output device, but does not explicitly disclose the output is a circular buffer. However, Cantrill in the same analogous art of a mechanism for ring buffering (circular buffer) in an arbitrary-action tracing framework (see for example, col.7, lines 15-17, "Embodiments of the invention provide a means for implementing a ring buffer scheme in arbitrary-action tracing frameworks which have variable length records."). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use circular buffer to store the output message. One would have been motivated to do so to keep the most recent recorded message in the fix sized buffer as suggested by Cantrill (col.1, lines 28-30, "one may which only want to keep the most recent data. To allow for this, tracing frameworks have historically implemented ring buffer.")

Claims 16, 26 and 36:

Claims 16, 26 and 36 are different product versions of method claim 6. It is well known in the computer that these products can be used to practice or perform the method discussed in claim 6 above. Therefore these claims are also unpatentable over Williams, GNU and PHP in view the teachings of Cantrill.

Conclusion

10. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Zheng Wei whose telephone number is (571) 270-1059 and Fax number is (571) 270-2059. The examiner can normally be reached on Monday-Thursday 8:00-15:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571- 272-1000.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ZW

ERIC B. KISS
PRIMARY EXAMINER